

Dồn bò [flowers]

Bò đi hái củi và để đàn bò gồm N con ăn cỏ trên cánh đồng như thường lệ. Khi quay lại, Bò phát hiện đàn bò đã vào vườn hoa và đang chén những cây hoa. Bò quyết định phải đưa lũ bò về chuồng ngay lập tức, hơn thế, còn phải dồn bò sao cho số hoa thiệt hại là ít nhất.

Con bò i đang ở địa điểm cách chuồng một khoảng T_i phút di chuyển, nó chén D_i cây hoa trong mỗi phút chờ tới lượt được đưa về chuồng. Ở mỗi thời điểm Bò chỉ có thể đưa một con bò về chuồng, với con bò i Bò sẽ mất $2 \times T_i$ phút (đưa bò về chuồng và quay lại).

Lập chương trình xác định số cây hoa bị ăn nếu Bò dồn bò theo lịch tối ưu.

Dữ liệu (flowers.inp)

- Dòng 1: ghi số nguyên N ($2 \leq N \leq 100,000$)
- Dòng 2 ... $N + 1$: mỗi dòng ghi hai số nguyên T_i ($1 \leq T_i \leq 2,000,000$) và D_i ($1 \leq D_i \leq 100$).

Kết quả (flowers.out)

- Dòng 1: số nguyên là số cây hoa bị ăn ít nhất.

Ví dụ

flowers.inp	flowers.out
6 3 1 2 5 2 3 3 2 4 1 1 6	86

Xếp thùng [stack]

Cho N thùng chứa đồ đánh số $1 \dots N$, thùng i có trọng lượng W_i và có thể chịu được tổng trọng lượng không quá C_i đặt trên. Hãy xác định số thùng nhiều nhất có thể xếp được thành một chồng.

Dữ liệu (stack.inp)

- Dòng 1: số nguyên N ($1 \leq N \leq 2,500$)
- Dòng 2: N số nguyên W_1, W_2, \dots, W_N ($1 \leq W_i \leq 100,000$)
- Dòng 3: N số nguyên C_1, C_2, \dots, C_N ($1 \leq C_i \leq 1,000,000,000$)

Kết quả (stack.out)

- Dòng 1: số nguyên là số thùng nhiều nhất có thể xếp thành một chồng

Ví dụ

stack.inp	stack.out
3 10 20 30 11 100 10	3
3 11 20 30 11 100 10	2

Đoàn thám hiểm (expedi)

Một đàn bò lấy một chiếc xe tải và liều lĩnh mở một cuộc thám hiểm vào sâu trong một khu rừng già. Thật không may, là những tài xế kém cỏi, đàn bò đã quyết định vượt qua một hòn đá bằng cách chạy vọt lên trên nó. Hậu quả là bình xăng của xe đã bị thủng. Chiếc xe bây giờ sẽ mất một đơn vị nhiên liệu cho mỗi đơn vị độ dài chúng phải đi.

Để sửa lại xe, những con bò phải đi đến thị trấn gần nhất trên một con đường dài và đầy gió. Trên con đường này, giữa thị trấn và vị trí hiện tại của xe có N trạm xăng. Chúng có thể dừng tại trạm xăng để đổ xăng.

Rừng già rất nguy hiểm cho con người, và càng đặc biệt nguy hiểm đối với bò. Do đó, chúng muốn dừng xe ít lần nhất trên đường đến thị trấn. Bình xăng của xe có kích thước coi như vô hạn. Chiếc xe hiện cách thị trấn L đơn vị độ dài và đang có sẵn P đơn vị xăng.

Xác định số lần ít nhất lũ bò phải dừng lại để có thể đến được thị trấn, hoặc chúng không thể thực hiện được việc này.

Dữ liệu (expedi.inp)

- Dòng 1: số nguyên N ($1 \leq N \leq 10,000$)
- Dòng 2 ... $N + 1$: mỗi dòng chứa hai số nguyên, được phân cách bởi khoảng trắng mô tả một trạm xăng. Số thứ nhất chỉ khoảng cách tính từ thị trấn, số thứ hai trong phạm vi 1 ... 100 chỉ lượng xăng có tại trạm đó.
- Dòng $N + 2$: hai số nguyên L, P ($1 \leq L, P \leq 1,000,000$).

Kết quả (expedi.out)

- Dòng 1: một số nguyên chỉ số trạm xăng cần phải ghé để có thể đến được thị trấn. Nếu không tồn tại khả năng này thì in ra -1 .

Ví dụ

expedi.inp	expedi.out
4 4 4 5 2 11 5 15 10 25 10	2

Giải thích

Ban đầu xe tải cách thị trấn 25 đơn vị độ dài, có 10 đơn vị xăng. Dọc con đường có 4 trạm xăng tại các mốc 4, 5, 11 và 15 tính từ thị trấn (tức là cách vị trí ban đầu của xe 21, 20, 14 và 10 đơn vị). Các trạm xăng cung cấp tối đa 4, 2, 5 và 10 đơn vị xăng theo thứ tự.

Cách đi tối ưu là: đi 10 đơn vị, lấy 10 đơn vị xăng, đi tiếp 4 đơn vị, lấy thêm 5 đơn vị xăng, sau đó đi thẳng đến thị trấn.

Sản xuất sữa chua #2 [yoghurt2]

Dioxin Milk là hãng sữa Mĩ nổi tiếng trên toàn thế giới, gần đây họ đầu tư dây chuyền sản xuất sữa chua. Các kĩ sư của hãng cho biết, hiện tại trong kho của nhà máy không còn hộp sữa chua

nào và trong N ngày tới, ngày thứ i hãng sản xuất được a_i hộp. Bộ phận bán hàng cho biết, đơn đặt hàng của N ngày tới tương ứng là b_i hộp cho ngày thứ i .

Vì mới tham gia vào thị trường sữa chua, Dioxin Milk cần thỏa mãn nhiều đơn hàng nhất có thể, khi đơn hàng cho một ngày nào đó bị từ chối, số hộp sữa còn lại của ngày hôm đó (bao gồm số sữa mới sản xuất trong ngày và số sữa còn tồn từ ngày hôm trước) được bảo quản cho ngày hôm sau.

Dựa trên thông tin sản xuất và thông tin về các đơn hàng, hãy xác định số đơn hàng nhiều nhất mà Dioxin Milk có thể đáp ứng và đưa ra một phương án phục vụ.

Dữ liệu (yoghurt2.inp)

- Dòng 1: số nguyên N ($1 \leq N \leq 250000$)
- Dòng 2: N số nguyên a_1, a_2, \dots, a_N ($0 \leq a_i \leq 10^9 \forall i$)
- Dòng 3: N số nguyên b_1, b_2, \dots, b_N ($0 \leq b_i \leq 10^9 \forall i$)

Kết quả (yoghurt2.out)

- Dòng 1: số nguyên K là số đơn hàng đáp ứng được nhiều nhất
- Dòng 2: K số nguyên theo thứ tự tăng là danh sách số thứ tự ngày của các đơn hàng được đáp ứng. Nếu có nhiều phương án thỏa mãn thì chỉ cần đưa ra một phương án bất kì.

Ví dụ

yoghurt2.inp

```
6
2 2 1 2 1 0
1 2 2 3 4 4
```

yoghurt2.out

```
3
1 2 4
```

Kéo xe [cardrag]

Bãi đỗ xe ô tô bị thu giữ vì vi phạm luật giao thông có dạng lưới ô vuông kích thước $M \times N$, mỗi xe nằm gọn trên một ô vuông theo một trong bốn hướng N, S, E, W (bắc, nam, đông, tây).

Vì một số nguyên nhân đặc biệt, quản lí bãi đỗ cần giải phóng nhanh chóng càng nhiều xe càng tốt. Muốn vậy cần đẩy các xe ra khỏi bãi mà không gây va chạm, tức là cần đẩy các xe theo hướng đỗ ban đầu đi ra khỏi bãi sao cho trong quá trình đẩy xe không đi vào ô vẫn còn xe. Một xe nếu đã bị di chuyển thì phải được kéo ra khỏi bãi.

Hãy xác định số lượng xe nhiều nhất có thể đẩy ra khỏi bãi với một trình tự đẩy xe tối ưu.

Dữ liệu

Gồm nhiều tests, mỗi test cho trên nhóm dòng theo định dạng:

- Dòng 1: hai số nguyên M, N ($1 \leq M, N \leq 2000$);
- Dòng 2 ... $M + 1$: mỗi dòng ghi xâu độ dài N chỉ gồm các kí tự trong tập $\{ . \ N \ S \ E \ W \}$ thể hiện ô trống hay ô có xe đỗ theo hướng tương ứng.

File dữ liệu kết thúc bằng dòng ghi hai số 0.

Kết quả

Kết quả mỗi test ghi trên một dòng số nguyên là số lượng xe nhiều nhất có thể đẩy ra khỏi bãi.

Ví dụ

cardrag.inp	cardrag.out
3 4 .N.W WWSS EWEW 0 0	4

Cân voi (scales)

Chán cảnh đi chăn trâu, Cuội bỏ lên miền núi chăn voi thuê cho nhà thống lí một bản nọ. Thống lí có một cô con gái rất xinh đẹp, lão "câu nhử" Cuội bằng cách giao hẹn nếu chăn đàn voi tốt thì sẽ gả con gái cho. Trời chiều lòng người, đàn voi Cuội chăn lớn nhanh như thổi. Giờ đây Cuội chỉ còn cân lũ voi nữa là ... xong.

Cuội được sử dụng một cân thăng bằng và N quả nặng đã biết khối lượng. Cuội cân voi bằng cách đưa voi lên một bàn cân và chất các quả nặng lên bàn cân bên kia cho đến khi thăng bằng. (Cuội không đưa quả nặng lên bàn cân có voi được vì lũ voi này rất hiếu chiến, chúng sẽ dùng vòi hất ngay các quả nặng đi). Chiếc cân cũng có giới hạn, nó sẽ gãy nếu Cuội đặt lên một trong hai bàn cân tổng khối lượng vượt quá C .

Các quả nặng có một tính chất khá thú vị là nếu sắp chúng thành một hàng theo khối lượng tăng dần thì mỗi quả (từ thứ ba trở đi) sẽ không nhẹ hơn tổng khối lượng hai quả liền trước. Để tránh khả năng bị thống lí "đánh tháo" vì làm gãy cân, Cuội muốn biết khối lượng lớn nhất có thể cân được là bao nhiêu. Hãy lập trình tính toán giúp Cuội.

Dữ liệu (scales.inp)

- Dòng 1: hai số nguyên N C ($1 \leq N \leq 1000$; $1 \leq C < 2^{30}$).
- Dòng 2 ... $N + 1$: mỗi dòng một số nguyên trong phạm vi số nguyên 31 bit có dấu là khối lượng của một quả nặng, theo thứ tự không giảm.

Kết quả (scales.out)

- Dòng 1: số nguyên chỉ khối lượng lớn nhất có thể cân được chính xác và an toàn.

Ví dụ

scales.inp	scales.out
3 15 1 10 20	11